Probabilistic Generative Models - Real NVP

Mykola LIASHUHA, Vaibhav ARORA

1. Introduction

In this work we implement real-valued non-volume preserving (real NVP). The goal is to find powerful, stably invertible, and learnable transformation, resulting in an unsupervised learning algorithm with exact log-likelihood computation, exact and efficient sampling, exact and efficient inference of latent variables, and an interpretable latent space [1] for a toy dataset.

2. Normalizing Flows

Normalizing flows introduced recently are used for density estimation or generative modeling. They have also been used for stochastic variational inference, as they provide an attractive approach for parameterizing flexible approximate posterior distributions in the VAE framework [2].

The typical approach for density estimation problems use generative probabilistic modeling (local models like VAEs or global models like RBMs). As data of interest are generally high-dimensional and highly structured, the challenge in this domain is building models that are powerful enough to capture its complexity yet still trainable. Their benefit compared to other models is that Normalizing Flow based models can perform efficient and exact inference, sampling and log-density estimation of data points [1].

Normalizing flows are based on the change of variable formula. Given a simple prior probability distribution p_Z on a latent variable $z \in Z$ and $f : X -> Z$ such that $g = f^{-1}$, we want to know how the distribution on X changes which can given by,

$$p_X(x) = p_Z(z)\left|det\left(\frac{\partial z}{\partial x}\right)\right|$$

$$\log(p_X(x)) = \log(p_Z(z)) + \log\left(\left|det\left(\frac{\partial z}{\partial x}\right)\right|\right)$$

where $\frac{\partial z}{\partial x}$ is the Jacobian of f at x.

A sample $z \sim p_Z$ is drawn in the latent space, and its inverse image $x = f^1(z) = g(z)$ generates a sample in the original space. Computing the density on a point x is accomplished by computing the density of its image $f(x)$ and multiplying by the associated Jacobian determinant $\left|det\left(\frac{\partial z}{\partial x}\right)\right|$ (see Figure 1). Exact and efficient inference enables the accurate and fast evaluation of the mode [1]. Normalizing Flows should satisfy several conditions in order to be practical [3]. They should:

- be invertible; for sampling we need g while for computing likelihood we need f
- be sufficiently expressive to model the distribution of interest
- be computationally efficient, both in terms of computing f and g (depending on the application) but also in terms of the calculation of the determinant of the Jacobian

3. Real NVP models

Computing the Jacobian of functions with high-dimensional domain and codomain and computing the determinants of large matrices are in general computationally very expensive. This combined with the restriction to invertible functions makes the problem appear impractical for modeling arbitrary distributions. But by careful design of the function f, a model can be learned which is both tractable and extremely flexible. As computing the Jacobian determinant of the transformation is crucial to effectively train using this principle, Real NVP exploits the simple observation that the determinant of a triangular matrix can be efficiently computed as the product of its diagonal terms [1].

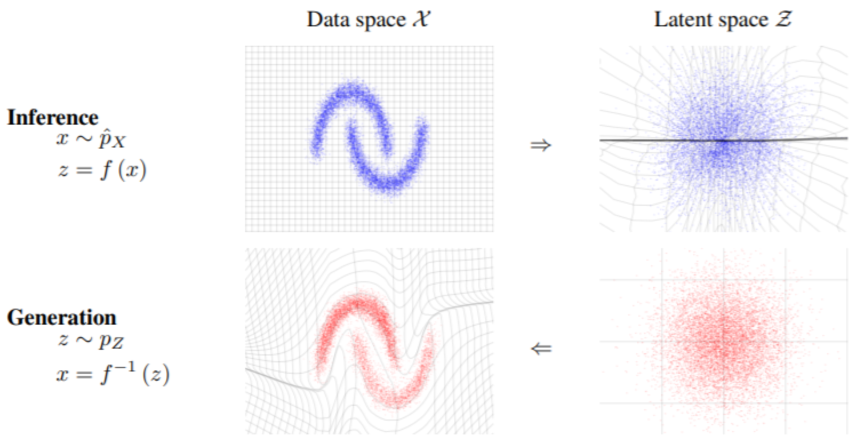


Figure 1: Real NVP learns an invertible, stable, mapping between a data distribution $\hat{p}_X$ and a latent distribution p_Z (typically a Gaussian). Here we show a mapping that has been learned on a toy 2-d dataset. The function $f(x)$ maps samples x from the data distribution in the upper left into approximate samples z from the latent distribution, in the upper right. This corresponds to exact inference of the latent state given the data. The inverse function, $f^{-1}(z)$, maps samples z from the latent distribution in the lower right into approximate samples x from the data distribution in the lower left. This corresponds to exact generation of samples from the model. The transformation of grid lines in $\mathcal{X}$ and $\mathcal{Z}$ space is additionally illustrated for both $f(x)$ and $f^{-1}(z)$.

Figure 1: Credits: [1]

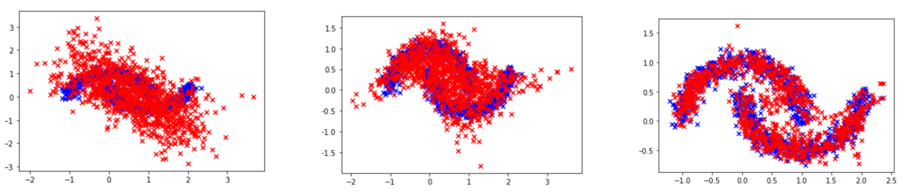
This can be achieved by stacking a sequence of simple bijections. In each simple bijection, part of the input vector is updated using a function which is simple to invert, but which depends on the remainder of the input vector in a complex way. Given a M dimensional input x and $m < M$, the output z of an affine coupling layer follows the equations,

$$z_{1:m} = x_{1:m}$$

$$z_{m+1:M} = x_{m+1:M} \odot \exp(s(x_{1:m})) + t(x_{1:m})$$

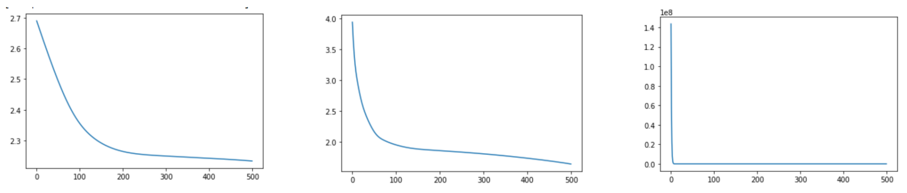
4. Results

It is intuitve that power of normalizing flow based models lies in the fact that multiple simple coupling layers can be stacked to increase expressivity of the model. Figure 2 shows the density estimation done on a toy example with different number of coupling layers. As we can observe, increasing the number of coupling layers allows to model the data better.



(a) 1 layer (b) 5 coupling layers (c) 50 coupling layers

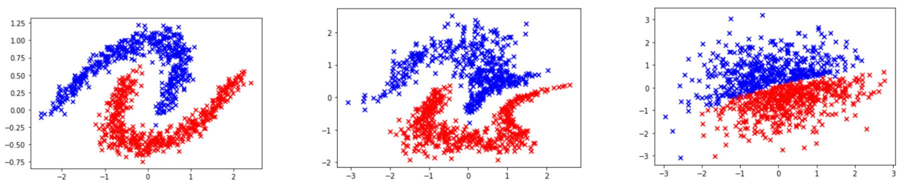
Figure 2: Density estimation with Real NVP



(a) 1 layer (b) 5 coupling layers (c) 50 coupling layers

Figure 3: Training loss

Moreover, in Figure 4, we display the latent space corresponding to each half moon of the toy dataset in a different color. We observe that with a single coupling layer, the data distribution to be modeled is almost simply mapped since a single layer is not expressive at all. But with 50 coupling layers, we do reach the desired gaussian latent space.



(a) 1 layer (b) 5 coupling layers (c) 50 coupling layers

Figure 4: Latent space

References

[1] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp, 2017. 1, 2

[2] D. P. Kingma. Variational inference deep learning: A new synthesis, 2017. 1

[3] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979, Nov 2021. 1